

# Compositional Learning of Relation Paths Embedding for Knowledge Base Completion

Xixun Lin<sup>1</sup>, Yanchun Liang<sup>1,2</sup>, Renchu Guan<sup>1,2\*</sup>

<sup>1</sup> Key Laboratory for Symbol Computation and Knowledge Engineering of National Education Ministry, College of Computer Science and Technology, Jilin University, Changchun 130012, China

<sup>2</sup> Zhuhai Laboratory of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Zhuhai College of Jilin University, Zhuhai 519041, China

## Abstract

Nowadays, large-scale knowledge bases containing billions of facts have reached impressive sizes; however, they are still far from completion. In addition, most existing methods only consider the direct links between entities, ignoring the vital impact about the semantic of relation paths. In this paper, we study the problem of how to better embed entities and relations into different low dimensional spaces. A compositional learning model of relation paths embedding (**RPE**) is proposed to take full advantage of additional semantics expressed by relation paths. More specifically, using corresponding projection matrices, RPE can simultaneously embed entities into corresponding relation and path spaces. It is also suggested that type constraints could be extended from traditional relation-specific to the new proposed path-specific ones. Both of the two type constraints can be seamlessly incorporated into RPE and decrease the errors in prediction. Experiments are conducted on the benchmark datasets and the proposed model achieves significant and consistent improvements compared with the state-of-the-art algorithms for knowledge base completion.

## 1 Introduction

Large-scale knowledge bases (KBs) such as Freebase (Bollacker et al., 2008), Yago (Suchanek et al., 2007), NELL (Carlson et al., 2010) are critical to natural language processing applications, e.g., question answering (Dong et al., 2015), relation extraction (Riedel et al., 2013), and language modeling (Ahn et al., 2016). These KBs usually contain billions of facts and each fact is organized as

the form of triple format (head entity, relation, tail entity), abbreviated as  $(h, r, t)$ , indicating that entities  $h$  and  $t$  hold the relationship  $r$ . These KBs are impressively large, however, their coverages are still far from completion compared with real-world knowledge (Dong et al., 2014). Traditional knowledge base completion approaches such as Markov logic networks (Richardson and Domingos, 2006) are suffered from the problems of feature sparsity and low-efficiency.

Recently, encoding the entire knowledge base into a low-dimensional vector space to learn latent representations for entities and relations has been attracting widespread attention. These knowledge embedding models yield better performance compared with prior work, in viewing of their low model complexity and high scalability. For example, TransE (Bordes et al., 2013) defines a score function  $S(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$  to measure the plausibility for triples. The score will get smaller if the triple  $(h, r, t)$  is more likely to be correct, otherwise it will get higher. The latent representations for entities and relations are learned by optimizing a global margin-loss function based on the total plausibility of observed triples. TransH (Wang et al., 2014) models each relation  $r$  as a hyperplane. Head entity  $h$  and tail entity  $t$  are projected by hyperplane norm vector to better tackle the attributes of relation, i.e., 1-to-1, 1-to-N, N-to-1, and N-to-N.

Both TransE and TransH suppose that entities and relations embeddings are in the same embedding space. TransR (Lin et al., 2015b) considers the entities from multiple aspects and various relations on different aspects. For each relation  $r$ , it defines a projection matrix, then the related entities are projected into relation-specific embedding space.

In spite of these knowledge embedding models are suitable to be deployed for large-scale KBs, most of them only exploit direct links connecting

\*Corresponding author: R.Guan (guanrenchu@jlu.edu.cn)

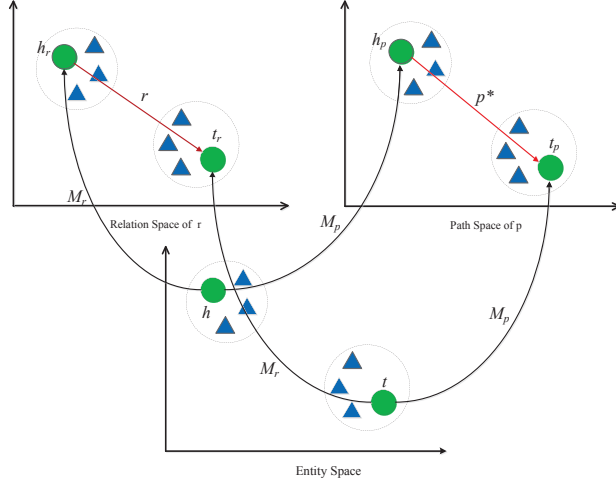


Figure 1: Simple illustration of relation-specific and path-specific projections. Each entity is projected by  $\mathbf{M}_r$  and  $\mathbf{M}_p$  into corresponding relation space and path space simultaneously. Different embedding spaces still hold the hypothesis:  $\mathbf{h}_r + \mathbf{r} \approx \mathbf{t}_r$ ,  $\mathbf{h}_p + \mathbf{p}^* \approx \mathbf{t}_p$ .  $\mathbf{p}^*$  denotes the relations composition for path representation, bold lower case letter  $\mathbf{v}$  denotes a column vector and bold upper case letter  $\mathbf{M}$  denotes a matrix.

head and tail entities to predict potential relations between entities. The new research direction using the semantic of relation paths to learn knowledge embeddings still needs to be explored (Lin et al., 2015a; Neelakantan et al., 2015; Guu et al., 2015; Toutanova et al., 2016).

In this paper, we propose a compositional learning model of relation paths embedding (RPE). The motivation of RPE is that the consistent semantics expressed by reliable relation paths should be similar to the semantic of relations between head and tail entities. Reliable relation paths are searched by path ranking algorithm (PRA) (Lao et al., 2011). Based on this motivation, we can extend the relation-specific projection and type constraints to our proposed path-specific ones. Figure 1 illustrates the basic idea for relation-specific and path-specific projections. We design two types of composition to construct path-specific projection  $\mathbf{M}_p$  dynamically without extra parameters, moreover, by means of a slight of changes on negative sampling, we also suggest that relation-specific and path-specific type constraints can be seamlessly incorporated into our model.

Our contributions can be summarized as: 1) To reinforce the complicated inference ability of

knowledge embedding models, path-specific projection is introduced. Then, the path-specific constraints can help to improve the model’s discriminability. In addition, compared with pure data-driven mechanism above knowledge embedding models used, the way that we utilize PRA to find out reliable relation paths can improve the knowledge representation learning interpretability. 2) In experiments, our model shows superior reasoning power to the prior work including TransE, TransH, TransR and PTransE in link prediction and triple classification tasks on benchmark datasets.

## 2 Related Work

In 2.1, we briefly introduce some previous works about classical translation-based models, i.e. (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015b), all of them exploit relation-specific projection to approach the challenge of attributes of relations, we also present the research on relation-specific type constraints (Krompass et al., 2015). And we introduce some works on relation path modeling (Lin et al., 2015a; Lao et al., 2011) in 2.2. Our model builds upon these works, extending the relation-specific projection and type constraints to path-specific ones for better latent representations of entities and relations.

### 2.1 Knowledge Embedding Models

We firstly review three translation-based models which only consider direct relations between entities. TransE considers that each entity or relation is a low dimensional vector in the same embedding space and the critical assumption for TransE is that every relation can be regarded as a translation from head entity to tail entity. For each triple  $(h, r, t)$ , TransE defines the score function as  $S(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|$ . Obviously, this assumption is simple and it can not deal with more complex relation attributes well, i.e., 1-to-N, N-to-1, and N-to-N.

To alleviate this problem, TransH proposes setting different hyperplane normal vector  $\mathbf{w}_r$  for different relation  $r$ .  $(h, t)$  are projected to the hyperplane, denoting as  $\mathbf{h}_h = \mathbf{h} - \mathbf{w}_r^T \mathbf{h} \mathbf{w}_r$ ,  $\mathbf{t}_h = \mathbf{t} - \mathbf{w}_r^T \mathbf{t} \mathbf{w}_r$  (Restrict  $\|\mathbf{w}_r\|_2 = 1$ ). The corresponding score function is  $S(h, r, t) = \|\mathbf{h}_h + \mathbf{r} - \mathbf{t}_h\|$ . Both TransE and TransH achieve translations in the same embedding space, TransR suggests each relation can be used to project entities into different relation-specific embedding spaces, in consideration of the

fact that different relations may emphasis on different entity aspects. The projected entity vectors are  $\mathbf{h}_r = \mathbf{M}_r \mathbf{h}$  and  $\mathbf{t}_r = \mathbf{M}_r \mathbf{t}$ ,  $\mathbf{M}_r$  is relation-specific projection matrix. The new score function is correspondingly defined as  $S(h, r, t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|$ .

Another research direction focuses on improving the performance of prediction by using the prior knowledge in format of relation-specific type constraints (Krompass et al., 2015; Chang et al., 2014; Wang et al., 2015). Notice that, each relation should possess *Domain* and *Range* fields to indicate the subject and object type respectively. For example, the relation *haschildren's* *Domain* and *Range* types both belong to person. By exploiting these limited rules, we can avoid the harmful influence of merely data-driven pattern, e.g., type-constrained TransE (Krompass et al., 2015) imposes these constraints on the global margin-loss function to better distinguish similar entities or relations embeddings in embedding space.

These representative knowledge embedding models are applicable to large-scale KBs. Unfortunately, all of them neglect the vital impact about the semantic of relation paths, which means that they are disabled for complicated reasoning scenes.

## 2.2 PTransE and PRA

Large-scale KBs are very huge heterogeneous directed graphs, composed of entities as nodes and relations as different types of edges. All entities constitute the entity set  $\zeta$ , all relations also constitute the relation set  $R$ . A relation path  $p$  can be represented as the sequence of relations, i.e.,  $p = (r_1, r_2, \dots, r_m)$ . In the case of concrete entity pair  $(h, t)$ , KBs usually contain redundant relation paths, i.e.,  $p_i$  ( $i=1, 2, \dots, n$ ). These relation paths have great significance for improving the power of inference for more complicate situations. For example, the sequences of triples already existed in KBs, (J. K. Rowling, CreatedRole, Harry Potter), (Harry Potter, Describedin, Harry Potter and the Philosophers Stone) can be used to infer the new fact (J.K. Rowling, WroteBook, Harry Potter and the Philosophers Stone) which does not appear in original KBs. However, not all paths are meaningful for our reasoning. Hence, PTransE uses path ranking algorithm (PRA) (Lao et al., 2011) to pick up reliable relation paths, more precisely, for each triple  $(h, r, t)$ ,  $P_{all} = \{p_1, p_2, \dots, p_k\}$  is the path

set for entity pair  $(h, t)$ . PRA calculates  $P(t|h, p_i)$ , the probability of reaching  $t$  from  $h$  following the sequences of relations indicated in  $p_i$ , which can be recursively defined as:

If  $p_i$  is an empty path:

$$P(t|h, p_i) = \begin{cases} 1 & \text{if } h = t \\ 0 & \text{else } h \neq t \end{cases} \quad (1)$$

If  $p_i$  is not an empty path,  $p'_i$  is defined as  $r_1, \dots, r_{m-1}$ , then

$$P(t|h, p_i) = \sum_{t' \in \text{Ran}(p'_i)} P(t'|h, p'_i) \cdot P(t|t', r_m) \quad (2)$$

$\text{Ran}(p'_i)$  is the set of target nodes where path  $p'_i$  ends at last.

PTransE obtains the reliable relation paths set  $P_{filter} = \{p_1, p_2, \dots, p_z\}$  by selecting relation paths which their probability above a certain threshold  $\eta$ . It also explores three different compositions of relations for path representation  $\mathbf{p}^*$ . Experiment results (Lin et al., 2015a) demonstrate the ADD composition i.e.,  $\mathbf{p}^* = \mathbf{r}_1 + \mathbf{r}_2 + \dots + \mathbf{r}_m$  achieves the best performance. PTransE defines the new score function as:

$$G(h, r, t) = S(h, r, t) + S(h, p, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| + \frac{1}{Z} \sum_{p_i \in P_{filter}} P(t|h, p_i) \cdot P_r(r|p_i) \cdot \|\mathbf{p}_i^* - \mathbf{r}\| \quad (3)$$

, where  $Z = \sum_{p_i \in P_{filter}} P(t|h, p_i)$  is normalization factor and  $P_r(r|p_i) = P_r(r, p_i) / P_r(p_i)$  is used to assist in calculating the relation paths reliance.

PRA, as one of the most promising research for knowledge base completion, also attracts much attention (Lao et al., 2015; Gardner et al., 2015; Wang et al., 2016; Nickel et al., 2016). It uses the path-constrained random walk probabilities  $P(t|h, p)$  as path features to train different linear classifiers for corresponding relations.

## 3 Our Model

All models mentioned above do not take full advantage of the semantic of relation paths, which has significant impact on learning meaningful latent representations for entities and relations. Here, we propose a compositional learning model of relation paths embedding (RPE), which includes novel path-specific projection and type constraints based on relation-specific ones. The consistent semantics expressed by relation paths are

used to better overcome the problem of relation’s mapping properties and improve the discriminability.

### 3.1 Path-specific Projection

The key idea of RPE is the consistent semantics expressed by reliable relation paths which filtered by PRA should be similar to the semantic of relation between entity pair. For a triple  $(h, r, t)$ , RPE exploits projection matrices  $\mathbf{M}_r, \mathbf{M}_p \in \mathbb{R}^{m \times n}$  to project entity pair’s vectors  $\mathbf{h}, \mathbf{t} \in \mathbb{R}^n$ , which exist in entity space, into corresponding relation space and path space simultaneously (m is the dimension of relation embeddings, n is the dimension of entity embeddings, m may differ from n). The projected vectors  $(\mathbf{h}_r, \mathbf{h}_p, \mathbf{t}_r, \mathbf{t}_p)$  in respective embedding spaces are denoted as :

$$\mathbf{h}_r = \mathbf{M}_r \mathbf{h}, \quad \mathbf{h}_p = \mathbf{M}_p \mathbf{h} \quad (4)$$

$$\mathbf{t}_r = \mathbf{M}_r \mathbf{t}, \quad \mathbf{t}_p = \mathbf{M}_p \mathbf{t} \quad (5)$$

In view of the fact that relation paths are sequences of relations  $p=(r_1, r_2, \dots, r_m)$ , we dynamically use the  $\mathbf{M}_r$  to construct  $\mathbf{M}_p$  for the purpose of decreasing model complexity. We explore two compositions for the formation of  $\mathbf{M}_p$ , which are formulated as:

$$\mathbf{M}_p = \mathbf{M}_{r_1} + \mathbf{M}_{r_2} + \dots + \mathbf{M}_{r_m} \quad (6)$$

(ADD Composition)

$$\mathbf{M}_p = \mathbf{M}_{r_1} * \mathbf{M}_{r_2} * \dots * \mathbf{M}_{r_m} \quad (7)$$

(MUL Composition)

ADD and MUL compositions respectively represent cumulative addition and cumulative multiplication, which are very simple and frequently be exploited in relevant works (GarcaDurn et al., 2015; Guu et al., 2015; Lin et al., 2015a). Matrix normalization is applied on  $\mathbf{M}_p$  for both compositions. In experiments, we evaluate these compositions’ performances on two subtasks of knowledge base completion (we named two compositions at PP-ADD and PP-MUL respectively). The new score function is defined as:

$$G(h, r, t) = S(h, r, t) + \lambda \cdot S(h, p, t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\| + \frac{\lambda}{Z} \sum_{p_i \in P_{filter}} P(t|h, p_i) \cdot P_r(r|p_i) \cdot \|\mathbf{h}_p + \mathbf{p}_i^* - \mathbf{t}_p\| \quad (8)$$

For path representation  $\mathbf{p}^*$ , we used the ADD composition, which as well shown in (Lin et al., 2015a;

GarcaDurn et al., 2015).  $\lambda$  is the hyper-parameter to balance the knowledge embedding score and relation path embedding score. In the experiments, we increase the limitation on these embeddings, i.e.,  $\|\mathbf{h}\|_2 \leq 1, \|\mathbf{t}\|_2 \leq 1, \|\mathbf{r}\|_2 \leq 1, \|\mathbf{h}_r\|_2 \leq 1, \|\mathbf{t}_r\|_2 \leq 1, \|\mathbf{h}_p\|_2 \leq 1, \|\mathbf{t}_p\|_2 \leq 1$ . By exploiting the additional semantic of relation paths, RPE improves its flexibility when modeling more complicate relation attributes. At the meantime, RPE can better tackle more complex reason scenes compared with prior works.

### 3.2 Path-specific Type Constraints

In RPE, based on the key idea, we can naturally extend the relation-specific to novel path-specific type constraints. In type-constrained TransE, the distribution of generating corrupted triples is uniform distribution.

In our model, we borrow the idea from (Wang et al., 2014), incorporating the two type constraints with Bernoulli distribution. For each relation  $r$ , we denote the  $Domain_r, Range_r$  to indicate the subject and object types of relation  $r$ .  $\zeta_{Domain_r}$  is the entity set whose entities are conformed to  $Domain_r$ ,  $\zeta_{Range_r}$  is the entity set whose entities are conformed to  $Range_r$ . We calculate the average numbers of tail entities for every head entity  $teh$  and the average numbers of head entities for every tail entity  $het$ . The Bernoulli distribution with parameter  $\frac{teh}{teh+het}$  for each relation  $r$  incorporated with two type constraints, which can be defined as: RPE samples entities from  $\zeta_{Domain_r}$  to replace the head entity with the probability  $\frac{teh}{teh+het}$ , and samples entities from  $\zeta_{Range_r}$  to replace the tail entity with the probability  $\frac{het}{teh+het}$ . The objective function for our model is defined as:

$$L = \sum_{(h,r,t) \in S} [L(h, r, t) + \frac{\lambda}{Z} \sum_{p_i \in P_{filter}} P(t|h, p_i) \cdot P_r(r|p_i) L(h, p_i, t)] \quad (9)$$

$L(h, r, t)$  is loss function for triples and  $L(h, p_i, t)$  is loss function for relation paths.

$$L(h, r, t) = \sum_{(h', r', t') \in S''} \max(0, S(h, r, t) + \gamma_1 - S(h', r, t')) \quad (10)$$

$$L(h, p_i, t) = \sum_{(h', r, t') \in S''} \max(0, S(h, p_i, t) + \gamma_2 - S(h', p_i, t')) \quad (11)$$

We denote  $S = \{(h_i, r_i, t_i) \mid i=1, 2, \dots, t\}$  as the set of all observed triples,  $S' = \{(h'_i, r_i, t_i) \cup (h_i, r_i, t'_i) \mid i=1, 2, \dots, t\}$  as the set of corrupted triples, each element of  $S'$  is obtained by randomly sampling from  $\zeta$ .  $S''$ , whose each element is conformed to the two type-constraints with Bernoulli distribution, is the subset of  $S'$ . The  $\text{Max}(0, x)$  returns the maximum between 0 and  $x$ .  $\gamma$  is the hyper-parameter of margin to separate corrected triples and corrupted triples. By exploiting the relation-specific and path-specific type constraints, RPE could better distinguish similar embeddings in different embedding spaces, so it can achieve better prediction quality.

### 3.3 Training Details

We adopted stochastic gradient descent (SGD) to minimize the objective function. We can exploit TransE or RPE (initial) for the initialization of entities and relations. We adapt score function of RPE (initial) as follows:

$$G(h, r, t) = S(h, r, t) + \lambda \cdot S(h, p, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\| + \frac{\lambda}{Z} \sum_{p_i \in P_{\text{filter}}} P(t|h, p_i) \cdot P_r(r|p_i) \cdot \|\mathbf{h} + \mathbf{p}_i^* - \mathbf{t}\| \quad (12)$$

, we also evaluate it in our experiment as our baseline. The projection matrices  $\mathbf{M}$  are initialized as identity matrices. RPE holds the local closed-world assumption (LCWA), each relation's domain and range types are based on the instance level. Their relation-specific type information are provided by KBs or the entities shown in our observed triples.

Notice that each relation  $r$  owns its reverse relation  $r^{-1}$ , therefore, to better learn the latent representations for entities and relations, RPE utilizes the reverse relation paths information. For example, the relation path *LeBron James*  $\xrightarrow{\text{PlayFor}}$  *Cleveland Cavaliers*  $\xrightarrow{\text{BelongTo}}$  *NBA*, its reverse relation path can be defined as *NBA*  $\xrightarrow{\text{BelongTo}^{-1}}$  *Cleveland Cavaliers*  $\xrightarrow{\text{PlayFor}^{-1}}$  *LeBron James*.

For every iteration we randomly sample a correct triple  $(h, r, t)$  with its reverse  $G(t, r^{-1}, h)$ , and the

Table 1: The statistics of datasets.

| Dataset | #Ent  | #Rel | #Train | #Valid | #Test |
|---------|-------|------|--------|--------|-------|
| FB15K   | 14591 | 1345 | 483142 | 50000  | 59071 |
| FB13    | 75043 | 13   | 316232 | 5908   | 23733 |
| WN11    | 38696 | 11   | 112581 | 2609   | 10544 |

final score function of our model is defined as:

$$F(h, r, t) = G(h, r, t) + G(t, r^{-1}, h) \quad (13)$$

In our implementation, we set the path length is 2 in consideration of numerating all relation paths are time-consuming. Moreover, as the path-constrained random walk probability  $P(t|h, p)$  suggests: with the increase of path length,  $P(t|h, p)$  will get smaller and more likely the relation path will be cast off.

## 4 Experiments

### 4.1 Datasets

We evaluate our model on two classical large-scale knowledge bases Freebase and WordNet. Freebase is a large collaborative knowledge base containing billions of facts about the real world, such as the triple (Beijing, Locatedin, China) describes the fact that Beijing is located in China. WordNet is a large lexical knowledge base of English, each entity is a synset expressing a distinct concept, and each relationship is conceptual-semantic or lexical relations. We use two subsets of Freebase: FB15K and FB13 (Bordes et al., 2013), one subset of WordNet: WN11 (Socher et al., 2013). Table 1 gives the statistics of the datasets.

In our model, each triple  $(h, r, t)$  has its own reverse triple  $(t, r^{-1}, h)$  to exploit the reverse relation paths information. Therefore, the total triples we actually used are as twice times as original dataset provides. Our model exploit the LCWA, both relation-specific and path-specific type information is based on the instance level. In this case, we utilize the type information provided by (Xie et al., 2016) for FB15K, as for FB13 and WN11, we do not depend on the auxiliary data, each relation's domain and range type information are approximated by triples original dataset contains.

Our model is evaluated in two subtasks of knowledge base completion: link prediction (Bordes et al., 2013) and triple classification (Socher et al., 2013).

## 4.2 Link Prediction

The task of link prediction is to predict the possible  $h$  or  $t$  for test triples  $(h, r, t)$  when  $h$  or  $t$  missing. We employ FB15K dataset for this task.

### 4.2.1 Evaluation Protocol

We follow the same evaluation procedures as used in (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015b). Firstly, for each test triple  $(h, r, t)$ , we replace  $h$  or  $t$  with every entities in  $\zeta$ . Secondly, each corrupted triple is calculated by corresponding score function  $S(h, r, t)$ . At last, we can rank of original correct entity with these scores in descending order.

Two metrics of evaluation are reported: the average rank of correct entities (Mean Rank) and the proportion of correct entities ranks in top 10 (Hits@10). Notice that if a corrupted triple already exists in knowledge base, it should not be considered as an incorrect one. We prefer to remove these corrupted triples from our dataset, and we call this setting as Filter. If these corrupted triples are reserved, we call this setting as Raw. In both settings, if the latent representations of entities and relations are better, the lower Mean Rank and the higher Hits@10 should be achieved.

### 4.2.2 Implementation

Because of the same dataset, we directly use the baseline results reported from (Lin et al., 2015b; Lin et al., 2015a) for comparison. We set the dimension of entity embedding  $m$  and relation embedding  $n$  among  $\{20, 50, 100, 120\}$ , the margin  $\gamma_1$  for calculate the score of  $S(h, r, t)$  among  $\{1, 2, 3, 4, 5\}$ , the margin  $\gamma_2$  for calculate the score of  $S(h, p, t)$  among  $\{3, 4, 5, 6, 7, 8\}$ , the learning rate  $\alpha$  for SGD among  $\{0.01, 0.005, 0.0025, 0.001, 0.0001\}$ , the batch size  $B$  among  $\{20, 120, 480, 960, 1440, 4800\}$ , the balance factor  $\lambda$  among  $\{0.5, 0.8, 1, 1.5, 2\}$ . We also set the threshold  $\eta$  in the range of  $\{0.01, 0.02, 0.04, 0.05\}$  in order to reduce the calculation of little sense of relation paths.

We use a grid search to determine the optimal parameters. The best configurations for RPE (PP-ADD) are  $n=100, m=100, \gamma_1=2, \gamma_2=5, \alpha=0.0001, B=4800, \lambda=1, \eta=0.05$ . We choose RPE (initial) to initialize our model and take  $L_1$  norm for score function, and we traverse our models for 500 epochs. Multi-thread training/testing is applied to learn entity and relation’s distributed representations.

Table 2: Evaluation results on link prediction

| Metric                                   | Mean Rank  |           | Hits@10(%)  |             |
|--|------------|-----------|-------------|-------------|
|  | Raw        | Filter    | Raw         | Filter      |
| TransE (Bordes et al., 2013)             | 243        | 125       | 34.9        | 47.1        |
| TransH (unif) (Wang et al., 2014)        | 211        | 84        | 42.5        | 58.5        |
| TransH (bern) (Wang et al., 2014)        | 212        | 87        | 45.7        | 64.4        |
| TransR (unif) (Lin et al., 2015b)        | 226        | 78        | 43.8        | 65.5        |
| TransR (bern) (Lin et al., 2015b)        | 198        | 77        | 48.2        | 68.7        |
| PTransE (ADD, 2-hop) (Lin et al., 2015a) | 200        | 54        | 51.8        | 83.4        |
| PTransE (MUL, 2-hop) (Lin et al., 2015a) | 216        | 67        | 47.4        | 77.7        |
| PTransE (ADD, 3-hop) (Lin et al., 2015a) | 207        | 58        | 51.4        | 84.6        |
| RPE (initial)                            | 207        | 58        | 50.8        | 82.2        |
| RPE (PC)                                 | 196        | 77        | 49.1        | 72.6        |
| RPE (PP-ADD)                             | <b>171</b> | <b>41</b> | 52.0        | <b>85.5</b> |
| RPE (PP-MUL)                             | 183        | 43        | <b>52.2</b> | 81.7        |
| RPE (PC + PP-ADD)                        | 184        | 42        | 51.1        | 84.2        |
| RPE (PC + PP-MUL)                        | 186        | 43        | 51.7        | 76.5        |

### 4.2.3 Results Analysis

Table 2 reports the results of link prediction,  $n$ -hop indicates the path length  $n$  PTransE exploits. We denote RPE only with path-specific constraints as RPE (PC), and from the results we can observe that: 1) Our models significantly outperform the classical knowledge embedding models (TransE, TransH, TransR) and PTransE on FB15K with the metrics of mean rank and hits@10. The results demonstrate that the path-specific projection and path-specific type constraints can explore more semantics about relation paths, which are crucial for large-scale knowledge base completion. 2) RPE (PC) improves little compared with baseline models. We think it is mainly caused that RPE (PC) only focuses on local information provided by related relations and entities, ignoring some global information compared with the way of randomly selecting corrupted entities. In mean rank, RPE (PP-ADD) achieves the best performance with 14.5% and 24.1% error reduction, compared with PTransE in raw and filter settings respectively. In hits@10, RPE (PP-ADD) brings few improvements. RPE with path-specific type constraints and projection is compromise between them.

Table 3 shows the evaluation results with separated types of relation properties on FB15K. From Table 3, we can conclude that 1) RPE (PP-ADD) outperforms all baselines in all mapping properties of relations. In particular, for the 1-to-N, N-to-1, N-to-N types of relations, which plague knowledge embedding models, RPE (PP-ADD) improves 5.3%, 6.0%, 4.9% compared with previous state-of-the-art performances respectively. 2) RPE (PP-MUL) performs worse than RPE (PP-ADD), and we think it is because RPE’s relations path composition is not consistent with RPE (PP-

Table 3: Evaluation results on FB15K by mapping properties of relations. (%)

| Tasks                                    | Predicting Head Entities (Hits@10) |             |             |             | Predicting Tail Entities(Hits@10) |             |             |             |
|--|------------------------------------|-------------|-------------|-------------|-----------------------------------|-------------|-------------|-------------|
|  | 1-to-1                             | 1-to-N      | N-to-1      | N-to-N      | 1-to-1                            | 1-to-N      | N-to-1      | N-to-N      |
| Relation Category                        |                                    |             |             |             |                                   |             |             |             |
| TransE (Bordes et al., 2013)             | 43.7                               | 65.7        | 18.2        | 47.2        | 43.7                              | 19.7        | 66.7        | 50.0        |
| TransH (unif) (Wang et al., 2014)        | 66.7                               | 81.7        | 30.2        | 57.4        | 63.7                              | 30.1        | 83.2        | 60.8        |
| TransH (bern) (Wang et al., 2014)        | 66.8                               | 87.6        | 28.7        | 64.5        | 65.5                              | 39.8        | 83.3        | 67.2        |
| TransR (unif) (Lin et al., 2015b)        | 76.9                               | 77.9        | 38.1        | 66.9        | 76.2                              | 38.4        | 76.2        | 69.1        |
| TransR (bern) (Lin et al., 2015b)        | 78.8                               | 89.2        | 34.1        | 69.2        | 79.2                              | 37.4        | 90.4        | 72.1        |
| PTransE (ADD, 2-hop) (Lin et al., 2015a) | 91.0                               | 92.8        | 60.9        | 83.8        | 91.2                              | 74.0        | 88.9        | 86.4        |
| PTransE (MUL, 2-hop) (Lin et al., 2015a) | 89.0                               | 86.8        | 57.6        | 79.8        | 87.8                              | 71.4        | 72.2        | 80.4        |
| PTransE (ADD, 3-hop) (Lin et al., 2015a) | 90.1                               | 92.0        | 58.7        | 86.1        | 90.7                              | 70.7        | 87.5        | 88.7        |
| RPE (initial)                            | 83.9                               | 93.6        | 60.1        | 78.2        | 82.2                              | 66.8        | 92.2        | 80.6        |
| RPE (PC)                                 | 82.6                               | 92.7        | 44.0        | 71.2        | 82.6                              | 64.6        | 81.2        | 75.8        |
| RPE (PP-ADD)                             | <b>92.5</b>                        | <b>96.6</b> | <b>63.7</b> | <b>87.9</b> | <b>92.5</b>                       | <b>79.1</b> | <b>95.1</b> | <b>90.8</b> |
| RPE (PP-MUL)                             | 91.2                               | 95.8        | 55.4        | 87.2        | 91.2                              | 66.3        | 94.2        | 89.9        |
| RPE (PC + PP-ADD)                        | 89.5                               | 94.3        | 63.2        | 84.2        | 89.1                              | 77.0        | 89.7        | 87.6        |
| RPE (PC + PP-MUL)                        | 89.3                               | 95.6        | 45.2        | 84.2        | 89.7                              | 62.8        | 94.1        | 87.7        |

MUL)’s composition of projection matrices for relation paths. Although RPE (PC) improves little compared with PTransE, we will indicate relation-specific and path-specific type constraints effectiveness in the task of triple classification. 3) We utilize the relation-specific projection matrix to construct path-specific projection matrix dynamically, and entities are encoded into relation space by relation-specific projection matrix and path space by path-specific projection matrix simultaneously. Experimental results demonstrate our model possesses better expressivity when modeling more complicate inference scenarios and mapping properties of relations.

### 4.3 Triple Classification

We conduct the task of triple classification on benchmark datasets to examine our model’s discriminative ability. Triple classification aims at predicting whether a given triple  $(h, r, t)$  is true.

#### 4.3.1 Evaluation Protocol

We set different relation-specific thresholds  $\{\delta_r\}$  to complete this task. For a test triple  $(h, r, t)$  if its score  $S(h, r, t)$  is below  $\delta_r$ , we predict it as a positive one, otherwise as a negative one.  $\{\delta_r\}$  is obtained by maximizing the classification accuracies on the valid set.

#### 4.3.2 Implementation

We compare our model with prior works, directly using the results about knowledge embedding models reported in (Lin et al., 2015b) for WN11 and FB13. Because (Lin et al., 2015a) does not evaluate PTransE’s performance on this task, we use the code of PTransE released in (Lin

Table 4: Evaluation results of triple classification. (%)

| Datasets                                 | WN11        | FB13        | FB15K       |
|--|-------------|-------------|-------------|
| TransE (unif) (Bordes et al., 2013)      | 75.9        | 70.9        | 77.8        |
| TransE (bern) (Bordes et al., 2013)      | 75.9        | 81.5        | 85.3        |
| TransH (unif) (Wang et al., 2014)        | 77.7        | 76.5        | 78.4        |
| TransH (bern) (Wang et al., 2014)        | 78.8        | 83.3        | 85.8        |
| TransR (unif) (Lin et al., 2015b)        | 85.5        | 74.7        | 79.2        |
| TransR (bern) (Lin et al., 2015b)        | 85.9        | 82.5        | 87.0        |
| PTransE (ADD, 2-hop) (Lin et al., 2015a) | 80.9        | 73.5        | 83.4        |
| PTransE (MUL, 2-hop) (Lin et al., 2015a) | 79.4        | 73.6        | 79.3        |
| PTransE (ADD, 3-hop) (Lin et al., 2015a) | 80.7        | 73.3        | 82.9        |
| RPE (initial)                            | 80.2        | 73.0        | 68.8        |
| RPE (PC)                                 | 83.8        | 77.4        | 77.9        |
| RPE (PP-ADD)                             | 84.7        | 80.9        | 85.4        |
| RPE (PP-MUL)                             | 83.6        | 76.2        | 85.1        |
| RPE (PC + PP-ADD)                        | <b>86.8</b> | <b>84.3</b> | <b>89.8</b> |
| RPE (PC + PP-MUL)                        | 85.7        | 83.0        | 87.5        |

et al., 2015a) to complete this part. FB13 and WN11 already contain negative samples, as for FB15K, we take the same process to produce negative samples randomly as (Socher et al., 2013) suggested instead of directly using the reported results. The hyper-parameter intervals are same as link prediction. The best configurations for RPE (PC + PP-ADD) are:  $n=50, m=50, \gamma_1=5, \gamma_2=6, \alpha=0.0001, B=1440, \lambda=0.8, \eta=0.05$ , taking  $L_1$  norm on WN11;  $n=100, m=100, \gamma_1=3, \gamma_2=6, \alpha=0.0001, B=960, \lambda=0.8, \eta=0.05$ , taking  $L_1$  norm on FB13;  $n=100, m=100, \gamma_1=4, \gamma_2=5, \alpha=0.0001, B=4800, \lambda=1, \eta=0.05$ , taking  $L_1$  norm on FB15K. We exploit RPE (initial) for initiation, epochs are limited in 500.

#### 4.3.3 Results Analysis

Table 4 lists the results for triple classification on different datasets. The results demonstrate that 1) RPE (PC + PP-ADD) achieves the best performance on all datasets, which takes good ad-

Table 5: Similar semantics expressed by relations and corresponding relation paths.

|                |   |
|----------------|---|
| entity pair    | (sociology, George Washington University)   |
| relation       | /education/field_of_study/students_majoring./education/education/institution  |
| relation paths | a: /education/field_of_study/students_majoring./education/education/student →<br>/people/person/education./education/education/institution<br>b: /people/person/education./education/education/major_field_of_study <sup>-1</sup> →<br>/education/educational_institution/students_graduates./education/education/student <sup>-1</sup> |
| entity pair    | (Planet of the Apes, art director)  |
| relation       | /education/field_of_study/students_majoring./education/education/institution  |
| relation paths | a: /film/film/sequel → /film/film_job/films_with_this_crew_job./film/film_crew_gig/film <sup>-1</sup><br>b: /film/film/prequel <sup>-1</sup> → /film/film/other_crew./film/film_crew_gig/film_crew_role   |

vantage of path-specific type constraints and path-specific projection; 2) RPE (PC) improves the performance of RPE (initial) with 4.5%, 6.0%, 13.2% especially on FB15K, so we consider that lengthening the distances for similar entities in embedding space is essential to specific problem. Experiment results also indicate that although LCWA can make up the loss for type information, real relation-type information is predominant.

#### 4.4 Case Study of semantic similarity

As shown in Table 5, we provide two relations and corresponding relation paths, which are considered as possessing similar semantics. They are extracted from Freebase and learned by RPE. For each example, we provide two relation paths, which their consistent semantics are actually similar to corresponding relations.

## 5 Conclusions and Future Work

In this paper, we propose a compositional learning model of relation paths embedding (RPE) for knowledge base completion. RPE’s key idea is making the consistent semantics expressed reliable relation paths similar to the relation between entity pair. To the best of our knowledge, it is the first time to propose the path-specific projection which utilizes the relation-specific projection matrices to build path-specific projection matrices dynamically. Moreover, it embeds the entities into relation space and path space to better deal with different type of relations simultaneously. We propose the novel path-specific type constraints based relation-specific ones to better distinguish similar entities in embedding space. In the future, we are going to 1) incorporate other potential semantic information into the relation paths modeling, such as the information provided by those intermediate

nodes connected by relation paths; 2) explore relation path embedding in other applications associated with knowledge bases, such as distant supervision for relation extraction and question answering over knowledge base.

## Acknowledgments

The authors are grateful for the support of the National Natural Science Foundation of China (No. 61572228, No.61272207, No.61472158, No.61300147) and the Science Technology Development Project from Jilin Province (20140520070JH, 20160101247JC).

## References

- [Bollacker et al.2008] Bollacker, Kurt and Evans, Colin and Paritosh, Praveen and Sturge, Tim and Taylor, Jamie 2008. *Freebase: a collaboratively created graph database for structuring human knowledge*. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pages 1247-1250.
- [Suchanek et al.2007] Suchanek, Fabian M and Kasneci, Gjergji and Weikum, Gerhard 2007. *Yago: a core of semantic knowledge*. In Proceedings of the 16th International Conference on World Wide Web, pages 697-706.
- [Carlson et al.2010] Andrew Carlson and Justin Beteridge and Bryan Kiesel and Burr Settles and Estevam R. Hruschka and Tom M. Mitchell 2010. *Toward an Architecture for Never-Ending Language Learning*. In Proceedings of the 24th AAAI Conference on Artificial Intelligence, pages 1306-1313.
- [Dong et al.2015] Li Dong and Furu Wei and Ming Zhou and Ke Xu 2015. *Question Answering over Freebase with Multi-Column Convolutional Neural Networks*. In Annual Meeting of the Association for Computational Linguistics (ACL).
- [Riedel et al.2013] Sebastian Riedel and Limin Yao and Andrew McCallum and Benjamin M. Marlin 2013.



- Relation Extraction with Matrix Factorization and Universal Schemas*. In NAACL, pages 74C84.
- [Ahn et al.2016] Sungjin Ahn and Heeyoul Choi and Tanel Pärnamaa and Yoshua Bengio 2016. *A Neural Knowledge Language Model*. CoRR, abs/1608.00318.
- [Bordes et al.2013] Antoine Bordes and Nicolas Usunier and Alberto García-Durán and Jason Weston and Oksana Yakhnenko 2013. *Translating Embeddings for Modeling Multi-relational Data*. In NIPS.
- [Wang et al.2014] Zhen Wang and Jianwen Zhang and Jianlin Feng and Zheng Chen 2014. *Knowledge Graph Embedding by Translating on Hyperplanes*. In AAAI.
- [Lin et al.2015b] Yankai Lin and Zhiyuan Liu and Maosong Sun and Yang Liu and Xuan Zhu 2015. *Learning Entity and Relation Embeddings for Knowledge Graph Completion*. In AAAI.
- [Lin et al.2015a] Yankai Lin and Zhiyuan Liu and Huan-Bo Luan and Maosong Sun and Siwei Rao and Song Liu 2015. *Modeling Relation Paths for Representation Learning of Knowledge Bases*. In EMNLP.
- [Neelakantan et al.2015] Arvind Neelakantan and Benjamin Roth and Andrew McCallum 2015. *Compositional Vector Space Models for Knowledge Base Completion*. In ACL.
- [Guu et al.2015] Kelvin Guu and John Miller and Percy Liang 2015. *Traversing Knowledge Graphs in Vector Space*. In EMNLP.
- [Toutanova et al.2016] Kristina Toutanova and Victoria Lin and Wen-tau Yih and Hoifung Poon and Chris Quirk 2016. *Compositional Learning of Embeddings for Relation Paths in Knowledge Base and Text*. In ACL.
- [Krompass et al.2015] Denis Krompass and Stephan Baier and Volker Tresp 2015. *Type-Constrained Representation Learning in Knowledge Graphs*. CoRR, abs/1508.02593.
- [Chang et al.2014] Kai-Wei Chang and Wen-tau Yih and Bishan Yang and Christopher Meek 2014. *Typed Tensor Decomposition of Knowledge Bases for Relation Extraction*. In EMNLP.
- [Wang et al.2015] Quan Wang and Bin Wang and Li Guo 2015. *Knowledge Base Completion Using Embeddings and Rules*. In IJCAI.
- [Lao et al.2011] Ni Lao and Tom M. Mitchell and William W. Cohen 2011. *Random Walk Inference and Learning in A Large Scale Knowledge Base*. In EMNLP.
- [Lao et al.2015] Ni Lao and Einat Minkov and William W. Cohen 2015. *Learning Relational Features with Backward Random Walks*. In ACL.
- [Gardner et al.2015] Matthew Gardner and Tom M. Mitchell 2015. *Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction*. In EMNLP.
- [Wang et al.2016] Quan Wang and Jing Liu and Yuanfei Luo and Bin Wang and Chin-Yew Lin 2016. *Knowledge Base Completion via Coupled Path Ranking*. In ACL.
- [Nickel et al.2016] Maximilian Nickel and Kevin Murphy and Volker Tresp and Evgeniy Gabrilovich 2016. *A Review of Relational Machine Learning for Knowledge Graphs*, volume 104. In Proceedings of the IEEE, pages 11-33.
- [García-Durán et al.2015] Alberto García-Durán and Antoine Bordes and Nicolas Usunier 2015. *Composing Relationships with Translations*. In EMNLP.
- [Socher et al.2013] Richard Socher and Danqi Chen and Christopher D. Manning and Andrew Y. Ng 2013. *Reasoning With Neural Tensor Networks for Knowledge Base Completion*. In NIPS.
- [Xie et al.2016] Ruobing Xie and Zhiyuan Liu and Maosong Sun 2016. *Representation Learning of Knowledge Graphs with Hierarchical Types*. In IJCAI.
- [Dong et al.2014] Xin Dong and Evgeniy Gabrilovich and Jeremy Heitz and Wilko Horn and Ni Lao and Kevin Murphy and Thomas Strohmann and Shaohua Sun and Wei Zhang 2014. *Knowledge vault: a web-scale approach to probabilistic knowledge fusion*. In KDD.
- [Richardson and Domingos2006] Richardson, Matthew and Domingos, Pedro 2006 *Markov logic networks* Machine Learning